



HAL
open science

How successfully is open-source research software adopted? Results and implications of surveying the users of a sensitivity analysis toolbox

Francesca Pianosi, Fanny J. Sarrazin, Thorsten Wagener

► To cite this version:

Francesca Pianosi, Fanny J. Sarrazin, Thorsten Wagener. How successfully is open-source research software adopted? Results and implications of surveying the users of a sensitivity analysis toolbox. *Environmental Modelling and Software*, 2020, 124, pp.104579. <10.1016/j.envsoft.2019.104579>. <hal-04754146>

HAL Id: hal-04754146

<https://hal.science/hal-04754146v1>

Submitted on 31 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC-ND 4.0 - Attribution - Non-commercial use - No Derivative Works - International License



Pianosi, F., Sarrazin, F., & Wagener, T. (2020). How successfully is open-source research software adopted? Results and implications of surveying the users of a sensitivity analysis toolbox. *Environmental Modelling and Software*, 124, Article 104579.
<https://doi.org/10.1016/j.envsoft.2019.104579>

Peer reviewed version

License (if available):
CC BY-NC-ND

Link to published version (if available):
[10.1016/j.envsoft.2019.104579](https://doi.org/10.1016/j.envsoft.2019.104579)

[Link to publication record on the Bristol Research Portal](#)
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via Elsevier at <https://www.sciencedirect.com/science/article/pii/S136481521930619X#!>. Please refer to any applicable terms of use of the publisher.

University of Bristol – Bristol Research Portal

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/brp-terms/>

How successfully is open-source research software adopted? Results and implications of surveying the users of a sensitivity analysis toolbox

Francesca Pianosi^{1,2,*}, Fanny Sarrazin^{1,3}, Thorsten Wagener^{1,2}

¹Department of Civil Engineering, University of Bristol, UK

²Cabot Institute, University of Bristol, UK

³Department of Computational Hydrosystems, UFZ-Helmholtz-Zentrum für Umweltforschung, Leipzig Germany

*Corresponding author: francesca.pianosi@bristol.ac.uk

Highlights

- Sharing open-access software has become common practice among researchers, but is rarely followed up with an analysis of its adoption success
- We were able to survey our toolbox users and found that workflows and commented code were effective in enabling users to adopt and tailor methods
- We also realized that effective uptake requires not only guidance for enabling users to produce results but also provision of guidance for their interpretation

Keywords

Research software; Open-source software; Reproducibility; Workflows; Software documentation

Abstract

Open-source research software is an important element of open science. While the number of software packages made available by researchers is increasing, there has been little analysis about their subsequent uptake. We collect basic information about prospective users when sharing our open-source sensitivity analysis toolbox. This enabled us to carry out a user survey to assess adoption success – beyond simply counting download numbers. Survey results confirm the key role of extensive documentation to ensure adoption, to enhance learning and to enable research implementation. We found that workflows are an effective tool to guide users to tailor methods to their problems. However, workflows also need to include guidance for interpretation of results, otherwise sophisticated functionalities are overlooked as their value is unclear. Developing effective documentation requires significant time investment but is essential if the ultimate aim of open research software is to promote the adoption of scientific methodologies and best practices.

1. Introduction

Sharing free, open-source research software is becoming increasingly common in the environmental modelling community. By ‘research’ software we refer here to software that is developed by researchers as part (but not as the primary focus) of their research activity – in contrast to ‘professional’ software produced by software engineering companies. Research software is now often distributed openly along with other key research outputs such as datasets, scientific papers or case study applications.

For example, limiting ourselves to the field of Sensitivity Analysis (Saltelli et al, 2008; Pianosi et al., 2016) to which the SAFE toolbox investigated in this paper contributes,

52 research software that has been made available over the years includes: the updated
53 version of the Simlab framework (Saltelli et al., 2008; JRC, 2015), the MCAT toolbox
54 (Wagener and Kollat, 2007), the GUI-HDMR software tool (Ziehn and Tomlin, 2009),
55 a new version (Poeter et al., 2014) of the UCODE for model calibration and local
56 sensitivity analysis (Poeter and Hill, 1999), a new release of the R Sensitivity package
57 (Iooss et al., 2018), the Python SALib library (Herman and Usher, 2017), and the
58 VARS-TOOL in Matlab/C++ (Razavi et al., 2019).

59

60 Motivations for freely sharing open-source software are manifold and have been
61 extensively investigated. Von Krogh et al. (2012) reviewed the ethical considerations
62 that historically inspired the Free Software movement (“*running software that a user*
63 *cannot inspect, modify and share is considered immoral*”), as well as the intrinsic and
64 extrinsic motivations that have driven developers thus far. Intrinsic motivations (i.e.
65 stemming from a pursuit of internal satisfaction rather than an external reward) include
66 the fun of developing software, a feeling of reciprocity (wherein one helps others
67 because of having been helped or expecting to be helped) and the gratification of
68 recognition by other community members (the “egoboo” drive; Raymond (1999)). We
69 would argue that all these intrinsic motivations apply to the case of research software
70 developers. An additional motivation can be the belief this activity serves science by
71 (1) increasing the uptake of our models and methods through the availability of the
72 software; and (2) by increasing the transparency and reproducibility of our own
73 analyses if the software is open-source (Stodden et al., 2016; Hutton et al., 2016;
74 Slater et al., 2019). External motivation can also be important, and many have
75 advocated for more explicitly rewarding research software development in academic
76 career progression. For example, new journals have been created for publishing
77 research software in the form of short, peer-reviewed and citable papers (examples
78 are *SoftwareX* and the *Journal of Open Source Software*), and specific funding
79 opportunities and research quality metrics have been advocated (Crouch et al., 2013).

80

81 We were motivated by all the above reasons when we released our open-source
82 “Sensitivity Analysis For Everybody” (SAFE) toolbox (Pianosi et al., 2015). We
83 particularly had the ambition to facilitate the use of Global Sensitivity Analysis (GSA)
84 by environmental modellers and to promote the uptake of what we considered best
85 practices in GSA application – reviewed in a companion paper (Pianosi et al. 2016).
86 These goals informed our design philosophy, which was characterised by (Pianosi et
87 al, 2015): (1) modularity of the code - in order to facilitate multi-method analyses; (2)
88 high density of comments – to facilitate understanding of the working principles of each
89 method; (3) minimal dependence on specific Matlab toolboxes/versions – to slow
90 down obsolescence; (4) availability of robustness and convergence metrics – to
91 enable rigorous analysis; (5) availability of visualization functions – to support
92 interpretation and communication of results; (6) availability of workflows - to help users
93 get started and to nudge them to follow best practices (enabled by point 1, 4 and 5).

94

95 The use of workflows, in place of a user manual, was one of the specific features of
96 SAFE. In general, a workflow is a series of connected steps employed to achieve an
97 overall goal (Duffy et al., 2012). In computational methods, workflows provide the
98 “information that explains what raw data and intermediate results are input to which
99 computations” (Stodden et al., 2016). In the SAFE toolbox, a workflow is an executable
100 script that shows, through a practical example, how the toolbox functions can be put
101 together to perform a sensitivity analysis (see an example in Figure 1). Users can

102 utilize workflows as tutorials to learn how to apply a specific method implemented in
103 SAFE. Or they can use workflows as a starting point to tailor the script for their own
104 application, given that many steps in GSA are common across problems and
105 applications. Encouraging users to produce and possibly share workflows is also an
106 effective way to promote transparency and reproducibility of the analyses. As such,
107 workflows can be seen as equivalent to the “modelling protocols” proposed by Ceola
108 et al. (2015) as a key mechanism to ensure comparability and reproducibility of model
109 comparison studies.

110

111 Since 2015, we have distributed SAFE through a dedicated website (safetoolbox.info)
112 while simultaneously collecting basic information about prospective users who
113 submitted a download request (such as their affiliation, area of expertise, etc.). In
114 2017, almost 3 years from the first software release and having received about 1000
115 download requests, we carried out a survey of SAFE users to evaluate its level of
116 adoption and the success of the design choices discussed above. In this paper, we
117 present the survey results to reflect on the efficacy of our approach to open-source
118 software design and distribution, and we draw general conclusions regarding the value
119 and effectiveness of different types of documentation, especially workflows.

120

121 The dataset of our survey respondents is quite unique. Indeed, many repositories used
122 to share research software (such as GitHub or Matlab central) do not collect either
123 contact details nor basic information of those who download the software. This makes
124 it very difficult to return to potential users for a survey, or to verify whether the sample
125 of respondents to a generic survey is representative of the overall population of
126 potential software users (given that the characteristics of that population would be
127 unknown) thus undermining the statistical significance of the survey results. Although
128 limited in scope, we thus think our results are interesting and provide some insights of
129 general interest to help other environmental modellers and software developers to
130 improve the quality and efficacy of their research software projects. We also hope that
131 they can inspire others to carry out similar surveys that could help tailor their efforts
132 for software development, distribution and uptake.

133

```

% Number of uncertain parameters subject to SA:
M = 5 ;
% Parameter ranges (from literature):
xmin = [ 0 0 0 0 0.1 ];
xmax = [400 2 1 0.1 1 ];
% Parameter distributions:
DistrFun = 'unif' ;
DistrPar = cell(M,1);
for i=1:M; DistrPar{i} = [ xmin(i) xmax(i) ] ; end
% Name of parameters (will be used to customize plots):
X_labels = {'Sm', 'beta', 'alfa', 'Rs', 'Rf'} ;

% Define output:
myfun = 'hymod_nse' ;

% Step 3 (sample inputs space)

r = 100 ; % Number of Elementary Effects
% [notice that the final number of model evaluations will be equal to
% r*(M+1)]

% option 1: use the sampling method originally proposed by Morris (1991):
% L = 6 ; % number of levels in the uniform grid
% design_type = 'trajectory'; % (note used here but required later)
% X = Morris_sampling(r,xmin,xmax,L); % (r*(M+1),M)

% option 2: Latin Hypercube sampling strategy
SampStrategy = 'lhs' ; % Latin Hypercube
design_type = 'radial';
% other options for design type:
% design_type = 'trajectory';
X = OAT_sampling(r,M,DistrFun,DistrPar,SampStrategy,design_type);

% Step 4 (run the model)
Y = model_evaluation(myfun,X) ; % size (r*(M+1),1)

% Step 5 (Computation of the Elementary effects)

% Compute Elementary Effects:
[ mi, sigma ] = EET_indices(r,xmin,xmax,X,Y,design_type);

```

Figure 1 – Example of workflow included in the SAFE toolbox. The workflow guides users in the application of a particular method, including explanatory comments and references. Users can execute the workflow ‘as is’ or change some of the input values and evaluate the impact of these choices. They can also use the workflow as a starting point to develop their own application, by changing the lines where their inputs or models differ from the example shown in the workflow.

134 2. Survey design

135

136 We divided the survey questionnaire into 3 main parts (the full survey – with responses
 137 - is included in the Supplementary Material):

138 Part 1 - general information about the respondent, including their expertise in GSA
139 and the extent to which they used SAFE (Q1-6).

140 Part 2 - specific information on the ways they used SAFE (Q7-14) aimed at testing
141 whether our design choices reached their intended goals.

142 Part 3 – some final questions about possible future directions for SAFE development
143 (Q15-18).

144 We used closed-answer questions in order to make the survey easier to complete (we
145 aimed at a response time of about 10 minutes) and to analyse. We included a N/A
146 (“Not Applicable”) answer option when needed in order to ensure that also
147 respondents who requested SAFE but did not actually use it were able to complete
148 the survey.

149 We used the Online surveys (formerly BOS) platform (www.onlinesurveys.ac.uk) to
150 distribute the survey to the 1000 researchers who had requested SAFE at the time
151 (November 2017). We kept the survey open for 10 days and sent two subsequent
152 reminders as the deadline was approaching. One useful feature of the Online surveys’
153 platform is that it can be set to send reminders only to those contacts that have not
154 responded yet.

155

156 **3. Survey results**

157

158 In this section, we report some key analyses of the survey responses. Given that, as
159 expected, only a fraction of survey recipients actually completed it, we first compare
160 some basic characteristics of the respondents’ sample and the surveyed population
161 (Sec. 3.1), such as their area of study/research and job title. The objective here is to
162 establish whether the sample of respondents provides a reasonable representation of
163 the overall population of SAFE users. We then move on to analyse the survey
164 responses (Sec 3.2), focusing on key points that may be of general interests to
165 research software developers (besides the GSA/SAFE user community). For the
166 interested reader, we report the full extent of the survey responses in the
167 Supplementary Material.

168

169 **3.1 Analysis of respondents**

170

171 We received n=195 responses from our surveyed population of N=1000 people who
172 requested SAFE over the period 2015-2017. This corresponds to a response rate of
173 almost 20% and a margin of error (at 95% confidence level) of:

174

175 margin of error (at 95% confidence level) = $\sqrt{\frac{N-n}{N-1} \frac{0.98}{\sqrt{n}}} 100 = 6.3\%$

176

177 (Isserlis, 1918; Sharon, 1999). Figure 2 analyses the basic characteristics of our
178 surveyed population (left) and of the survey respondents (right). The top panels show
179 that the areas of study/research are well represented in the respondents’ sample (i.e.
180 their relative extent is similar in the two populations). The bottom panels instead show
181 some differences in terms of roles, as Bachelor/Master students are underrepresented
182 in the respondents’ sample, and lecturers/professors are slightly overrepresented. We
183 assume this may be due to the fact that Bachelor/Master students are more likely to
184 disengage from the research area after completing their student projects; some of
185 them may have also not received the survey at all if they lost access to the University
186 email account after graduation.

187
188
189
190

Despite these small differences, we believe the respondents sample to be an acceptable representation of the target population.

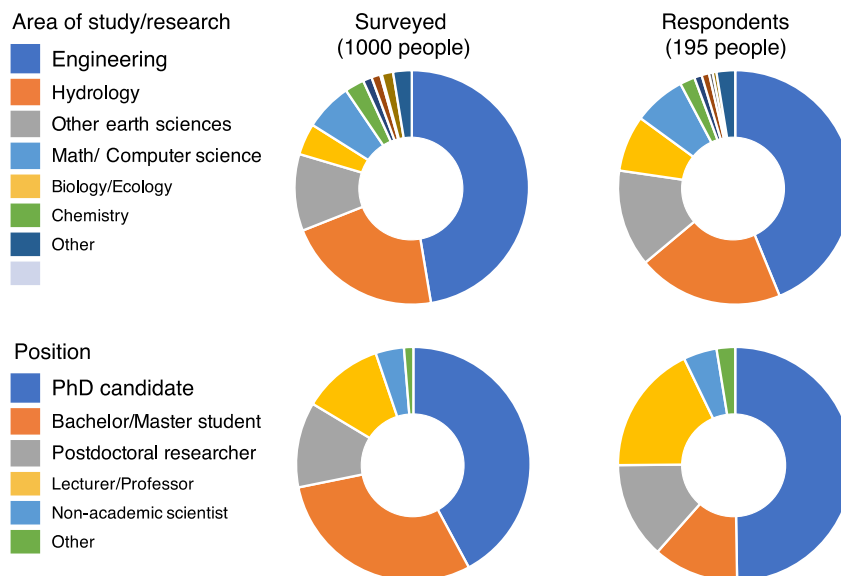


Figure 2 – Area of study/research (top) and role (bottom) of the 1000 people we surveyed as they requested SAFE between 2015 and 2017 (left) and of the 195 people who completed our survey (right).

191
192
193
194
195
196
197
198

3.2 Analysis of responses

Figure 3, 4 and 5 show a selection of the survey answers and the key points we can draw from them. Beyond specific feedback on SAFE that will be useful for us to improve the toolbox, we believe that the following points of general interest arise.

The availability of open-source software is attractive to users with diverse levels of expertise (among our respondents, 22.1% said they had “no expertise at all” of GSA when they requested SAFE, 56.9% had “basic” expertise, and 21% had “good/advanced” expertise – see Q4 in the Supplementary Material and the top panel of Figure 3). This is in line with our expectations and design choices, which were meant to make the toolbox useful for both experienced and novice users (Pianosi et al., 2015).

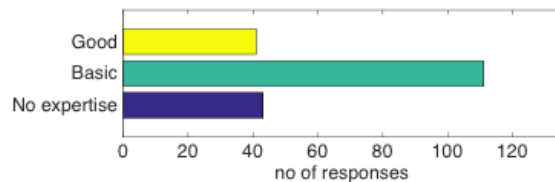
Roughly half of the people who requested the software did actually use it. About a fifth (17.8%) instead did not even try it and a third (30.3%) tried it but did not find it useful for their research (question Q3 and middle panel in Fig. 3). This result suggests that, somehow expectedly, the number of download requests of a free software may significantly overestimate the number of actual users.

On the other hand, the respondents who did use the toolbox largely benefitted from it, both through producing useful results (for 29.7% of all respondents up to the point of including them into a publication; Q3) and through improving their understanding of the underpinning GSA methods (43.6% said their understanding increased “somewhat” and 36.4% said it increased “significantly”; Q5 in Fig. 3). This result

217

218 confirms that sharing open-source software is an effective way to also promote the
 219 understanding and uptake of methodologies by a wider range of researchers.
 220 Interestingly, the increased understanding seems to be equally perceived by users
 221 who started with no expertise in GSA as well as by those who already had a good
 222 initial level of understanding.
 223
 224

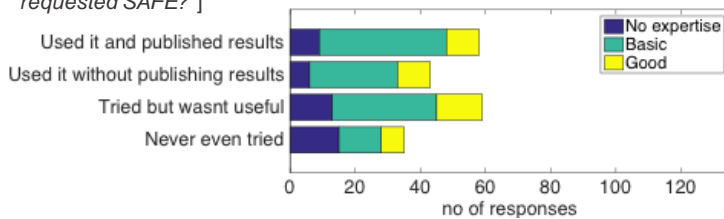
Q4: How would you judge your level of expertise in Global Sensitivity Analysis (GSA) when you requested SAFE?



Key points

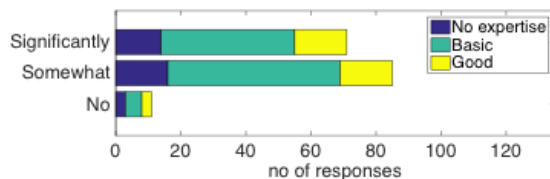
SAFE was requested by researchers with a very diverse level of expertise in GSA, from no expertise (coloured in blue in all bar plots) to good/advanced (yellow).

Q3: How extensively did you use SAFE? [split according to response to: "Q4: How would you judge your level of expertise in GSA when you requested SAFE?"]



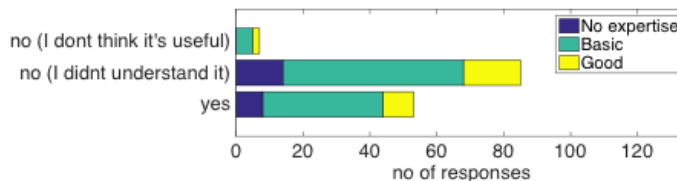
Of those who requested a copy of SAFE, about half (52%) actually used it. Interestingly, about a third of the respondents not only used it but also got to publish results.

Q5: Do you think using SAFE increased your understanding of GSA? [same]



Additionally, slightly more than a third said using SAFE increased their understanding of GSA significantly. This seems to apply equally across the (declared) level of expertise in GSA at the time of requesting SAFE.

Q11: Did you ever use the bootstrapping functionalities to look at robustness of sensitivity indices? [same]



On a less positive end, almost half of respondents said they did not use the bootstrapping functionalities available in SAFE, despite our efforts to highlight their value for a rigorous GSA application.

Figure 3 – Analysis of the responses to selected questions of the survey about the use and usefulness of the SAFE toolbox and some of its advanced functionalities such as bootstrapping (note: plots of responses to Q5 and Q11 do not include the "N/A" answers by respondents who did not actually use SAFE; full responses to all questions are given in the Supplementary Material).

225
 226 Some users exploited the functionalities of SAFE in line with our suggested best
 227 practice, for example by applying multiple GSA methods (29.2%; Q9 in the
 228 Supplementary Material) and by complementing the analyses with visualisation
 229 functions (42.6%; Q10). However, more sophisticated functionalities such as
 230 bootstrapping, which we consider essential to ensure the trustworthiness of
 231 conclusions drawn from GSA results (Sarrazin et al., 2016), was not picked up as
 232 much as we hoped. Surprisingly to us, 43.6% of respondents declared they did not
 233 use it because they did not understand it, and 3.6% declared they do not think it is

234 useful at all (Q11 and bottom panel in Figure 3). This result suggests that, unless a
 235 specific effort is made in highlighting the value of more sophisticated functionalities,
 236 many users are likely to overlook them.
 237

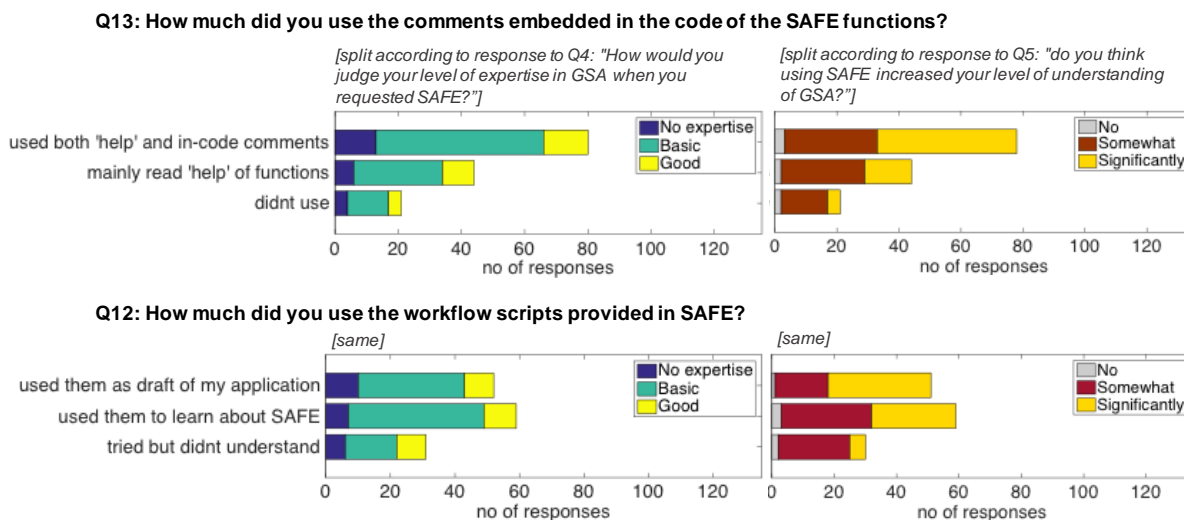


Figure 4 – Analysis of the responses to selected questions of the survey about the effectiveness of the SAFE toolbox documentation (note: plots do not include the “N/A” answers by respondents who did not actually use SAFE; full responses to all questions are given in the Supplementary Material).

238
 239
 240 Most users read and appreciate comments, both in the form of function help (22.6%)
 241 and in-code comments within functions (41%) (Q13 and Figure 4). Again, the
 242 repartition of responses according to level of initial expertise is quite uniform (top left
 243 panel in Figure 4), which suggests that users with different backgrounds found the
 244 documentation equally accessible. However, the top right panel in Figure 4 shows that
 245 the users who mostly engaged with the in-code documentation are more likely to
 246 perceive a significant increase in their understanding of GSA. This result suggests that
 247 sharing well commented open-source software is an effective mechanism to improve
 248 understanding of methodologies.

249
 250 Workflow scripts also proved useful (Q12 and bottom panel of Figure 4): a third of
 251 respondents used them to learn about specific GSA methods and about a fourth
 252 (26.7%) employed them as an initial draft for their own workflow. Similar to the in-code
 253 comments, we see here as well that the users who engaged most with the workflows
 254 also found their general learning of GSA improved most significantly.

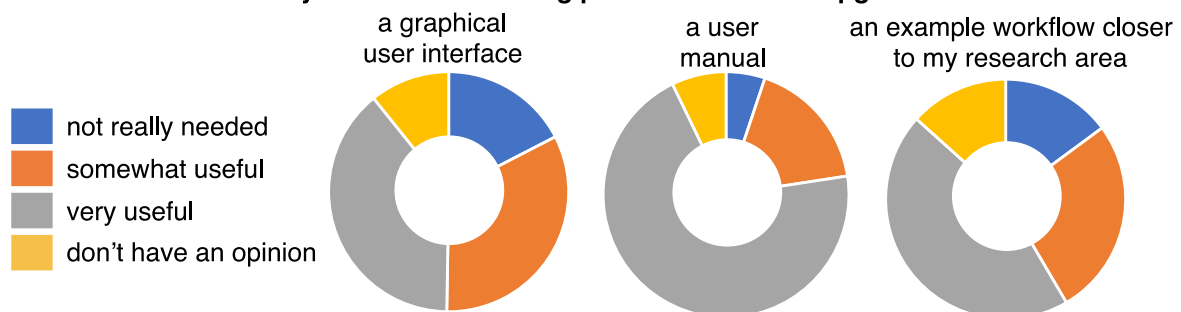
255
 256 When asked about possible future developments, many respondents suggested
 257 workflows closer to their application area (45.1%) (see Q15 in the Supplementary
 258 Material and Figure 5), and the majority said they would also want a user manual
 259 (70.3% of respondents said it would be “very useful”). In our approach to developing
 260 documentation, we assumed that the scientific background knowledge provided by a
 261 manual could be conveyed equally effectively by peer-reviewed papers that we
 262 published along with the SAFE toolbox. For example, our literature review paper
 263 (Pianosi et al., 2016) includes an extensive description of the available methods in

264 SAFE as well as the key steps in the set-up of GSA, which are mirrored one-to-one in
 265 the SAFE workflows. However, we acknowledge that we must have failed in clearly
 266 communicating the availability of such scientific documentation. Indeed the literature
 267 review and other scientific papers (such as Sarrazin et al (2016), which provides the
 268 background to the bootstrapping functionalities in SAFE) are linked in the FAQ page
 269 of the SAFE Toolbox website (www.safetoolbox.info/faq/), which 31.3% of our
 270 respondents said they did not realize was available (see Q14 in the Supplementary
 271 Material). In order to better highlight these resources we have now modified the email
 272 message by which the toolbox is sent to users.

273
 274 Lastly, in responding to Q14 and/or to the final open-ended question, a good number
 275 of users said that they would want a version of SAFE in R or Python. This gave us
 276 motivation to accelerate our plans to develop such versions, which have now been
 277 made available through the same SAFE website.

278

Q15 How useful would you find the following potential additions/upgrades to SAFE?



279
 280 *Figure 5 – Analysis of the responses to selected questions of the survey (full*
 281 *responses in the Supplementary Material) about possible additions to the SAFE*
 282 *toolbox.*

283

284

4 Outlook

285

286

287 In 2015 we started sharing an open-source toolbox for Global Sensitivity Analysis
 288 (GSA). We chose to distribute the toolbox through a bespoke website with an
 289 electronic registration form, which enabled us to collect some basic information from
 290 everybody downloading the software. This enabled us to perform a survey that proved
 291 very useful for us to measure the actual uptake of the software and the effectiveness
 292 of some of our design choices, as well as to inform upgrades to the toolbox. On the
 293 other hand, we are aware that this choice may be frowned upon by some potential
 294 users, and it reduces the interaction with the user base and the integration of their
 295 contributions – which would both be easier if the code was hosted on a public platform
 296 such as GitHub. Given the relatively specialised nature of the toolbox, and hence the
 297 relatively small size of its user community, it is still possible for us to manage
 298 interactions via email, and we have occasionally uploaded add-ons contributed by
 299 users on the website FAQ page¹. While we recognise this approach may become

¹ Some figures: we had received 1000 download requests at the time of the survey (November 2017) and about 2200 at the time of writing this manuscript (September 2019). We currently receive about one email a month with request for additional clarification or comments from users. We know some users have tailored the toolbox functions to their specific needs (see for example the comments to our survey's open-ended question in the Supplementary material) but they have rarely shared this back

300 cumbersome when dealing with larger communities or frequent software updates, thus
301 far we have found that the advantages, such as enabling user surveys, outweigh its
302 limitations.

303

304 We decided to share our toolbox mainly as a way to increase the impact and utilization
305 of GSA methods. The survey results presented in this paper seem to confirm that
306 software availability has helped other researchers in their applications of GSA - in
307 about 30% of cases all the way to the point of publishing the results. Importantly, the
308 majority of users also declared that using the toolbox increased their understanding of
309 the methods it makes available. We believe that the key to this success lies in the
310 extensive documentation we developed (in the form of function helps, in-code
311 comments, workflows, and a website with links to application examples and Frequently
312 Asked Questions). The survey results further confirm that documentation is essential
313 to both enable the use of the software as well as to realise its potential for increasing
314 the understanding of the underpinning methods.

315

316 However, developing a extensive documentation requires time and resources that go
317 well beyond what is needed for the development of the code itself. Such effort was
318 acceptable within the context of the research project in which SAFE was first
319 developed (the CREDIBLE project; NE/J017450/1). In fact, developing software tools
320 was regarded as one of the ways to achieve the broader project goal of improving the
321 consideration of model uncertainty in natural hazard assessment. In other cases,
322 researchers may share their code for different aims. For example, some may 'just'
323 want to avoid duplication of efforts and save other researchers' the time needed to
324 carry out similar developments. In this case, it may be unreasonable to expect
325 researchers to also develop thorough documentation when they are already sharing
326 their work for free. The point we would like to make here is that setting a clear goal for
327 distributing open-access research software determines the amount of effort put into
328 developing the documentation that is needed to reach that goal.

329

330 A key lesson learnt from the survey is that, if software is meant to facilitate the uptake
331 of a methodology and its appropriate application, users need to be supported not only
332 in generating results but also in interpreting them. Such a need has been openly
333 recognised by our survey respondents (for example, in response to the final open-
334 ended question, one user said: "*A user guide featuring basic approaches to*
335 *interpreting results could be very useful*") and is implicitly suggested by the limited
336 uptake of those functionalities, such as bootstrapping, whose meaning is not
337 immediately self-evident. So, documentation should not be limited to enabling users
338 to produce results but should also provide guidance on interpreting them and for
339 understanding their implications. Users will not adopt methods if they do not
340 understand the value of the information they provide, and they will fall back on the
341 simplest and easiest to understand software functionalities. The survey results also
342 confirm that workflows are an effective tool for knowledge transfer. Though, in order
343 to achieve our objectives fully, workflows should include both guidance on how to
344 produce results as well as how to interpret them. We have hence developed additional
345 documentation and workflows specifically focused on analyzing and interpreting the

with us. So far, we have updated the Matlab toolbox once, and sent the new release to users by email. We recently added the option to download an R and a Python version of the toolbox and we are planning another update of the Matlab version.

346 meaning and implications of key set-up choices in GSA application (Wagener and
347 Pianosi, 2019; Noacco et al., 2019).

348
349 An exciting opportunity for the implementation of user-friendly workflows comes in the
350 form of interactive notebooks that are becoming increasingly easy to develop, thanks
351 to new tools such as the R Shiny package (<https://shiny.rstudio.com>), the Wolfram
352 Notebook Interface for Mathematica (<http://www.wolfram.com/mathematica/>) and
353 Jupiter notebooks for Python (<https://jupyter.org>). We believe that these new packages
354 offer an unprecedented opportunity for developing interactive workflows that are
355 extremely effective in supporting users to explore methods, choices and results in
356 general. Besides supporting knowledge transfer and training, workflows are also
357 highly valuable for increasing the transparency and reproducibility of individual
358 applications (Hutton et al., 2016). Shared workflows (often connected to published
359 journal papers as supplemental material) enable other users to reproduce previous
360 analyses and provide a starting point for users to develop their own applications – thus
361 directly benefiting from previous software/method tailoring. Ultimately, shared
362 workflows provide an agile mechanism to increase the transparency and
363 reproducibility of computational research.

364

365 **Acknowledgement**

366

367 The SAFE toolbox was initially supported by the UK Natural Environment Research
368 Council (NERC) through the Consortium on Risk in the Environment: Diagnostics,
369 Integration, Benchmarking, Learning and Elicitation (CREDIBLE) [NE/J017450/1]. F.
370 Pianosi is currently supported by the UK Engineering and Physical Sciences Research
371 Council (EPSRC) through a “Living with Environmental Uncertainty” fellowship
372 [EP/R007330/1]. The authors wish to thank three anonymous reviewers for their
373 comments that helped improving the discussion of the survey results. The authors are
374 also grateful to the many users of the SAFE toolbox who over the years provided their
375 feedbacks and comments through individual contacts and by taking part in the survey
376 presented in this paper.

377

378 **References**

379

380 Ceola, S., Arheimer, B., Baratti, E., Blöschl, G., Capell, R., Castellarin, A., Freer, J.,
381 Han, D., Hrachowitz, M., Hundecha, Y., Hutton, C., Lindström, G., Montanari, A.,
382 Nijzink, R., Parajka, J., Toth, E., Viglione, A., and Wagener, T. (2015): Virtual
383 laboratories: new opportunities for collaborative water science, *Hydrol. Earth Syst.*
384 *Sci.*, 19, 2101-2117.

385

386 Crouch, S., Chue Hong, N., Hettrick, S., Jackson, M., Pawlik, A., Sufi, S., Carr, L., De
387 Roure, D., Goble, C., Parsons, M. (2013), *The Software Sustainability Institute:*
388 *Changing Research Software Attitudes and Practices*, *Computing in Science &*
389 *Engineering*, 15(6), 74-80.

390

391 Duffy, C. J., Gil, Y., Deelman, E., Marru, S., Pierce, M., Demir, I., & Wiener, G. (2012).
392 Designing a road map for geoscience workflows. *Eos*, 93(24), 225-226.

393

394 Herman, J. and W. Usher (2017), SALib: An open-source Python library for Sensitivity
395 Analysis, *The Journal of Open Source Software*, 2(9).

396
397 Hutton, C., T. Wagener, J. Freer, D. Han, C. Duffy, and B. Arheimer (2016). Most
398 computational hydrology is not reproducible, so is it really science? *Water Resour.*
399 *Res.*, 52, 7548-7555.
400
401 looss, B., Janon, A. and Pujol, G. (2018), Sensitivity: Global Sensitivity Analysis of
402 Model Outputs, <https://CRAN.R-project.org/package=sensitivity> (last visited: 18 April
403 2019)
404
405 Isserlis, L. (1918), On the value of a mean as calculated from a sample. *Journal of the*
406 *Royal Statistical Society*, 81(1): 75-81.
407
408 Joint Research Center (2015), SIMLAB and other software,
409 <https://ec.europa.eu/jrc/en/samo/simlab> (last visited: 18 April 2019)
410
411 Krogh, G. V., Haefliger, S., Spaeth, S. and Wallin, M. W. (2012). Carrots and
412 Rainbows: Motivation and Social Practice in Open Source Software Development.
413 *MIS Quarterly*, 36(2), pp. 649-676.
414
415 Lohr, S. L. (1999), *Sampling: Design and Analysis*. Pacific Grove, California: Duxbury
416 Press.
417
418 Noacco V., Sarrazin, F., Pianosi, F., Wagener, T. (2019), Matlab/R workflows to
419 assess critical choices in Global Sensitivity Analysis using the SAFE toolbox,
420 *MethodsX*, 6, 2258-2280.
421
422 Pianosi, F., F. Sarrazin, and T. Wagener (2015), A Matlab toolbox for global sensitivity
423 analysis, *Environmental Modeling & Software*, 70, 80-85.
424
425 Pianosi F., Beven, K., Freer, J., Hall, J.W., Rougier, J., Stephenson, D.B. and
426 Wagener, T. 2016. Sensitivity analysis of environmental models: A systematic review
427 with practical workflow. *Environmental Modeling & Software*, 79, 214-232.
428
429 Poeter, E.P. and Hill, M.C. (1999), UCODE, a computer code for universal inverse
430 modeling, *Computers & Geosciences*, 25(4), 457-462.
431
432 Poeter, E.P., Hill, M., Lu, D., Tiedeman, C., Mehl, S. (2014) UCODE_2014, with New
433 Capabilities To Define Parameters Unique to Predictions, Calculate Weights Using
434 Simulated Values, Estimate Parameters with SVD, Evaluate Uncertainty with MCMC,
435 and More. GWMI 2014-02. Integrated Groundwater Modeling Center
436
437 Raymond, E. 1999. *The Cathedral and the Bazaar: Musings on Linux and Open*
438 *Source by an Accidental Revolutionary*, revised ed. O'Reilly & Associates, Boston,
439 MA.
440
441 Razavi, A., Sheikholeslami, R., Gupta, H.V., Haghnegahdar, A. (2019), VARS-TOOL:
442 A toolbox for comprehensive, efficient, and robust sensitivity and uncertainty analysis,
443 *Environmental Modelling & Software*, 112, 95-107.
444

445 Saltelli, A., M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana,
446 S. Tarantola, (2008), *Global Sensitivity Analysis. The Primer*. Wiley.
447
448 Sarrazin, F., Pianosi, F. and Wagener, T. 2016. Global Sensitivity Analysis of
449 environmental models: Convergence and validation. *Environmental Modeling &
450 Software*, 79, 135-152.
451
452 Slater, L.J., Thirel, G., Harrigan, S., Delaigue, O., Hurley, A., Khouakhi, A., Prodocimi,
453 I., Vitolo, C., Smith, K. (2019), Using R in hydrology: a review of recent developments
454 and future directions, *Hydrology and Earth System Science Discussion*, in review,
455 2019.
456
457 Stodden, V., McNutt, M., Bailey, D. H., Deelman, E., Gil, Y., Hanson, B., Heroux, M.A.,
458 Ioannidis, J.P.A. and Taufer, M. (2016), Enhancing reproducibility for computational
459 methods, *Science*, 354(6317), 1240-1241.
460
461 Wagener, T. and Kollat, J. (2007). Numerical and visual evaluation of hydrological and
462 environmental models using the Monte Carlo analysis toolbox. *Environmental
463 Modelling & Software*, 22(7), 1021-1033.
464
465 Wagener, T. and Pianosi, F. (2019). What has Global Sensitivity Analysis ever done
466 for us? A systematic review to support scientific advancement and to inform policy-
467 making in earth system modelling. *Earth-Science Reviews*, 194, 1-18.
468 doi.org/10.1016/j.earscirev.2019.04.006
469
470 Ziehn, T., Tomlin, A., 2009. GUI-HDMR - a software tool for global sensitivity
471 analysis of complex models. *Environmental Modeling & Software*. 24 (7), 775-785.